

# Basic metrics for requirements management

---

A Borland White Paper

*By Chris Kolde, Senior Software Architect/Technical Lead, National Accounts*

March 2004

---

**Borland<sup>®</sup>**

## Contents

Introduction.....	3
Measuring the requirements process .....	3
Measures for the project manager .....	4
Number of requirements by owner/responsible person.....	5
Number of requirements by status/total number of requirements .....	5
Functional requirements allocated to a project release or iteration .....	5
Requirements growth over time .....	6
Number of requirements completed.....	6
Number of requirements traced or not traced.....	6
Measuring the benefits of requirements management.....	6

## Introduction

This paper presents a simple set of requirements-related metrics that projects can adopt when adding requirements management practices to their existing development process, or as part of a broader effort to improve the process of eliciting, documenting, and managing requirements throughout the software lifecycle to support organizational goals or attain various certifications.

The identified metrics are not meant to represent a comprehensive set. Rather, they represent a simple set of measurements that projects new to requirements management can choose from, depending on their needs. Regardless of the software process being followed, every project documents requirements in some form using a variety of artifacts. However, many projects lack even the simplest requirements-related measurements to help manage the project to successful completion, avoid rework, control scope, or manage change during the project. The identified metrics can be applied to virtually any software development effort.

The identified metrics fall into three categories. First, there are metrics that help assess the goodness of the requirements process itself. Second, there are metrics that provide the project manager and project leaders with objective information to help guide the project to successful completion. Finally, there are metrics to help assess the impact requirements management is having on overall project costs and product quality. Some of these metrics can be gathered from requirements management tools; others need to be gathered from the tools used for managing change requests or tracking defects.

## Measuring the requirements process

Changes to the requirements of a system should be expected and encouraged early in the lifecycle as the stakeholders and development team reach a common understanding of what the system should do. However, excessive changes to the requirements, especially later in the lifecycle, can lead to project failure. Sometimes the failure is spectacular; millions of dollars are spent on a project that is ultimately cancelled. Other cases are less extreme, and perhaps

result only in schedule slippage, reduced functionality, customer dissatisfaction, or lost business opportunities.

The following requirements measure the amount of change on a project and whether those changes are related to the requirements. Excessive requirements-related change will require corrective action and may be an indicator of a broken requirements process.

- Frequency of change in the total requirements set
- Rate of introduction of new requirements
- Number of requirements changes to a requirements baseline
- Percentage of defects with requirement errors as the root cause
- Number of requirements-related change requests (as opposed to defects found in testing or inspections)

## Measures for the project manager

There are several metrics the project manager may use to get an objective measure of the state of a project and, if necessary, take corrective action. Alternatively, the metrics may indicate the project is ahead of plan and may be able to deliver more business value than originally anticipated. While the metrics below are readily extracted if the members of the project team (analysts, developers, testers, project manager, etc.) are updating information about the requirements using a requirements management tool, the metrics need to be interpreted within the context of the project and where it is in its lifecycle.

## Number of requirements by owner/responsible person

These metrics indicate the workload of various people on the project. The information can be used by the project manager to determine whether the project could benefit by shifting some of the workload. It can also be used to determine whether the right people are assigned to specifying or implementing the most important requirements.

## Number of requirements by status/total number of requirements

The set of requirements for a project are constantly in flux, especially during the earlier phases of development. Some requirements may have been approved and will be incorporated into the product being developed. Others may have been proposed by one or more stakeholders, but there is not yet agreement about whether they will be included in the product. Other requirements will be in various stages of development (e.g., being worked on, coding is complete, validated by testing, completed). Still others may be on hold pending clarification of certain issues. Having a clear understanding of exactly what the state of each requirement is and where it is in the development process enables the project manager to effectively manage the project, avoid requirements and scope creep, and take corrective actions to deliver the project on time and within budget while assuring that all the critical business needs are satisfied.

## Functional requirements allocated to a project release or iteration

Understanding exactly how many requirements, and which specific ones, are allocated to a release or iteration allows the project manager to successfully deliver the project on time with the most critical functionality. Making this information available to the team keeps everyone focused. It can also shorten development cycles by allowing the QA team to get an early start with test planning, test development, and establishing the appropriate test environment, while the code is being developed.

## Requirements growth over time

Early in the lifecycle, this metric can help the project manager determine whether adequate progress is being made gathering and specifying the requirements. As the project progresses, unusual growth can be an indicator of scope creep. It may also be an indicator that there are opportunities to improve the way in which requirements are elicited and documented.

## Number of requirements completed

This is an objective indicator of the number of requirements implemented, tested, and validated to date. Trends of requirements completed over time can also help measure how quickly the project is moving toward completion (e.g., the velocity) and whether the project team can include more functionality or is potentially overcommitted.

## Number of requirements traced or not traced

Many projects adopt various levels of formal requirements traceability to help ensure the completeness of the system and understand the impact on other requirements, designs, code, and tests should the requirements change. Understanding this impact can help the project better understand the cost of proposed changes and control the scope of the project so it can be successful within its cost and schedule constraints. If traceability is being adopted on a project, understanding which requirements are or are not traced can be useful indicators of progress and completeness. For example, at certain points in the lifecycle, the number of critical business requirements not yet traced to designs and code could indicate a need for corrective action.

## Measuring the benefits of requirements management

Several independent studies confirm that requirement errors are the most frequent project errors. These errors precipitate defects in architecture, design, and implementation. If the resulting software errors are not detected during testing, they most certainly will be detected

post-launch, and there business impact could be severe. In either case, they lead to costly changes for the project and can result in scrapping or reworking significant parts of the application. Good requirements management, as part of an overall requirements process, can reduce the number of defects, reduce project costs pre- and post-launch, and improve the overall quality of the product. The following metrics can be indicators of the benefits from requirements management. The first two metrics, when combined with other measures, can be used to calculate a monetary return.

- Trend of post-launch defects over time
- Trend of number of change requests for rework – both pre-launch and post-launch
- Customer satisfaction surveys

**Made in Borland®** Copyright © 2004 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Corporate Headquarters: 100 Enterprise Way, Scotts Valley, CA 95066-3249 • 831-431-1000 • [www.borland.com](http://www.borland.com) • Offices in: Australia, Brazil, Canada, China, Czech Republic, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Italy, Japan, Korea, Mexico, the Netherlands, New Zealand, Russia, Singapore, Spain, Sweden, Taiwan, the United Kingdom, and the United States. • 21858